Stackelberg strategies for transportation networks

Walid Krichene   Jack Reilly

Abstract

We study inefficiencies of transportation networks due to the selfish behaviour and lack of coordination of drivers, by comparing social optimal equilibria to Nash equilibria. Then we investigate possible strategies to reduce the inefficiency by studying the Stackelberg routing game: assuming we have control over a fraction of the flow on the network, what is a good way of routing that compliant flow so that the induced Nash equilibrium is closer to the social optimum than the initial Nash equilibrium? Stackelberg scheduling on parallel link networks has been studied in a non-transportation setting, and it is shown in [1] that computing the optimal Stackelberg assignment is NP-hard in the number of links for increasing latency functions. We first describe the problem in the specific setting of transportation networks, where the dynamics of flow result in latency functions that do not satisfy common properties assumed in the literature studying Stackelberg scheduling. We then characterize Nash equilibria for our network and show that there are multiple such equilibria. Then we study the Stackelberg game and describe optimal Stackelberg strategy in the case of 2 link networks, and show results for general parallel link networks.

1 The Model

1.1 Traffic flows

We consider a network of N parallel links indexed by $n \in \{1, \ldots, N\}$, under constant positive flow demand, or rate r. The flow x_n on link n is a function of the density ρ_n, given by a triangular fundamental diagram with the following parameters

- v_n the free-flow speed on the link

- w_n the congestion wave speed

- $x_n^{\max}$ the maximum capacity of the link

In the free flow regime (when the density on the link is less than a critical density ρ_n^c that is given by $v_n \rho_n^c = x_n^{\max}$) the velocity is constant and the flow increases linearly in the density $x_n = v_n \rho_n$. In the congested regime ($\rho_n^c < \rho_n \leq \rho_n^{\max}$ the density on the link is greater than the critical density and less than a maximum density given by $x_n^{\max} - w_n(\rho_n^{\max} - \rho_n^c) = 0$), the flow decreases linearly in the density $x_n = x_n^{\max} - w_n(\rho_n - \rho_n^c) = -w_n(\rho_n - \rho_n^{\max})$ and the velocity is decreasing.

$$x_n = \begin{cases} v_n \rho_n & \rho_n \leq \rho_n^c \\ x_n^{\max} - w_n(\rho_n - \rho_n^c) = -w_n(\rho_n - \rho_n^{\max}) & \rho_n^c < \rho_n \leq \rho_n^{\max} \end{cases} \tag{1}$$

Note that the flow can also be written succinctly as

$$x_n = \min\{v_n \rho_n, -w_n(\rho_n - \rho_n^{max})\} \tag{2}$$

We denote by (N, r) a network instance with N links, rate r, and no compliant flow. We next define feasible flow assignments for network instance (N, r).

Definition 1. *A flow assignment $x \in \mathbb{R}_+^N$ is feasible for instance (N, r) if $\forall n \; x_n \leq x_n^{\max}$ and $\sum_n x_n = r$*

If x is a feasible flow assignment for (N, r), we denote by $Supp(x)$ the support of x, that is the set of links that are used by the flow assignment

$$Supp(x) = \{n | x_n > 0\}$$

1.2 Steady state equilibria

We are interested in the steady state equilibria of the network under constant positive rate r. In the steady state equilibria the flow and density variables are static.

1.3 Latency function

The velocity on link n is given by x_n/ρ_n, and the individual latency function is

$$l_n(\rho_n, x_n) = \frac{L_n \rho_n}{x_n}$$

where L_n is the length of link n.

Using the expression 1 of flow as a function of the density, we can write

$$l_n(\rho_n) = \begin{cases} \frac{L_n}{v_n} & \rho_n \leq \rho_n^c \\ \frac{L_n}{w_n(\rho_n^{\max}/\rho_n - 1)} & \rho_n^c < \rho_n \leq \rho_n^{\max} \end{cases}$$

The total latency incurred by all users on a link is $x_n l_n(\rho_n) = L_n \rho_n$, and the total latency incurred by all users on the network is

$$C(\rho) = \sum_n L_n \rho_n \tag{3}$$

1.3.1 Latency as a function of flow

We can express the latency as a function of flow by introducing an integer $m_n \in \{0, 1\}$ that specifies whether link n is congested ($m_n = 1$ if n is congested and $m_n = 0$ if n is in free-flow)

$$\rho_n(x_n, m_n) = \begin{cases} \frac{x_n}{v_n} & m_n = 0 \\ \frac{1}{w_n}(x_n^{\max} - x_n) + \rho_n^c & m_n = 1 \end{cases}$$

this corresponds to inverting the fundamental diagram that gives the flow as a function of density. The latency is then given by

$$l_n(x_n, m_n) = \frac{L_n \rho_n}{x_n} = \begin{cases} \frac{L_n}{v_n} & m_n = 0 \\ L_n \left(\frac{\rho_n^{\max}}{x_n} - \frac{1}{w_n} \right) & m_n = 1 \end{cases}$$

For convenience of notation, we introduce constants a_n and b_n such that

$$l_n(x_n, m_n) = \begin{cases} a_n & m_n = 0 \\ b_n \left(\frac{1}{x_n} - \frac{1}{x_n^{\max}} \right) + a_n & m_n = 1 \end{cases} \tag{4}$$

where $a_n = \frac{L_n}{v_n}$ is the free-flow latency and $b_n = L_n \rho_n^{\max}$.

1.3.2 Properties of the latency function

The latency function does not satisfy properties usually assumed in the Stackelberg scheduling literature. In particular, the latency is not an increasing function of flow: it is a constant function if the link is in free-flow, and a *decreasing* function when the link is congested.

And for a given flow x_n, there are up to two possible latencies, one corresponding to the free-flow regime (few cars on the link moving fast) and one to the congested regime (many cars on the link moving slowly).

As a consequence, some of the known results on congestion networks do not apply to our setting: for instance, the network has multiple Nash equilibria that have different costs. In the next section we specify this result and address the issue of having multiple pure Nash equilibria.

2 Nash Equilibria

In this section we characterize pure Nash equilibria of the network, which we simply refer to as Nash equilibria. We first review the essential uniqueness of Nash equilibria in the case of increasing latency functions (in the sense that all Nash equilibria have the same cost). Then we show that latency functions consistent with the fundamental diagram of traffic induce multiple Nash equilibria with different costs, and we analytically solve for Nash equilibria for $(N = 2, r)$ instances, and give some results for $(N > 2, r)$ instances.

2.1 Characterization of Nash equilibria

Definition 2. *An assignment $(q, m) \in \mathbb{R}_+^N \times \{0, 1\}^N$ for instance (N, r) is at Nash equilibrium, if $\forall n$*

$$x_n > 0 \Rightarrow \forall k, l_n(x_n, m_n) \leq l_k(q_k, m_k)$$

In particular, every (non-atomic) user cannot improve her latency by switching to another link. As a consequence, all links that are in the support of q have the same latency l_0, and links that are not in the support have latency greater than l_0. Note that to fully characterize the equilibrium one needs to specify the congestion state m, since the latency on a link depends on whether the link is congested.

Lemma 1. *If (q, m) is an assignment for instance (N, r) at Nash equilibrium, then*

$$x_n > 0 \Rightarrow l_n(x_n, m_n) = l_0$$
$$x_n = 0 \Rightarrow l_n(0, 0) \geq l_0$$

and the total latency incurred by the network is $C(q, m) = rl_0$.

Note that links that have zero flow are necessarily in free-flow $x_n = 0 \Rightarrow m_n = 0$.

2.2 Nash equilibria for increasing latency functions

Assuming the latency functions $x_n \to l_n(x_n)$ are increasing, one can show that all Nash equilibria have the same cost. Let x and x' be two assignments for (N, r) at Nash equilibrium. Let l_0, respectively l_0' denote the common latency of all links in the support of x, respectively x'. The cost of the Nash equilibria are respectively rl_0 and rl_0'.

Assume $x \neq x'$. Then $\exists n_1, n_2$ such that

$$x_{n_1} > x'_{n_1} \geq 0 \qquad\qquad x'_{n_2} > x_{n_2} \geq 0$$

Since q is at Nash equilibrium and $n_1 \in Supp(x)$, $l_{n_1}(x_{n_1}) \leq l_{n_2}(x_{n_2})$. And since l_{n_2} is increasing $l_{n_2}(x_{n_2}) \leq l_{n_2}(x'_{n_2})$. Thus $l_0 = l_{n_1}(x_{n_1}) \leq l_{n_2}(x_{n_2}) \leq l_{n_2}(x'_{n_2}) = l_0'$. Exchanging the roles of x and x' we have $l_0' \leq l_0$. Therefore $l_0 = l_0'$ and both equilibria have the same cost.

2.3 Traffic networks have multiple Nash equilibria

To simplify our discussion, we assume, without loss of generality, that the links are ordered by increasing free-flow latencies

$$a_1 \leq a_2 \leq \cdots \leq a_N$$

2.4 Nash equilibria for 2-parallel link networks

We consider a $(2, r)$ instance. We characterize the Nash equilibria of the network depending on the flow demand r.

2.4.1 Equal free-flow latencies

2.4.2 Different free-flow latencies

2.5 Nash equilibria for general parallel link networks

We consider a (N, r) instance where $N \geq 2$. We further assume that the free-flow latencies are different to avoid degenerate cases where the set of Nash equilibria is infinite

$$a_1 < a_2 < \cdots < a_N$$

Under this assumption, we show that there are a least 2 Nash equilibria, one purely congested, and one where there is a single link in free flow, and show that there are at most N equilibria.

3 Social Optimal equilibria

Consider an instance (N, r) where the flow demand r does not exceed the maximum capacity of the network $r \leq \sum_n x_n^{\max}$. The social optimal steady state equilibrium of the network is a solution to the following optimization problem (SO)

$$\min_{\rho, q} \sum_n L_n \rho_n$$
$$\text{subject to } \sum_n x_n = r$$
$$x_n = \min\{v_n \rho_n, w_n(\rho_n^{\max} - \rho_n)\}$$

This problem is non convex due to the second flow constraint $x_n = f(\rho_n) = \min\{v_n \rho_n, x_n^{\max} - w_n(\rho_n - \rho_n^c)\}$ which corresponds to the fundamental diagram of traffic. However, we show that the solutions to this optimization problem are necessarily in free-flow ($x_n = v_n \rho_n \ \forall n$, or equivalently, $m = 0$), thus the social optimum can be computed by solving an equivalent linear program.

3.1 Social Optima are in free-flow

Lemma 2. (ρ^*, x^*) is optimal for (SO) only if $x_n^* = v_n \rho_n^* \ \forall n$

Proof: Let (ρ, q) be a feasible point for (SO), and suppose that $\exists k$ such that $x_k < v_k \rho_k$. We show that $\exists$ feasible point $(\bar{\rho}, \bar{x})$ such that $C(\bar{\rho}) < C(\rho)$. Let

$$\bar{x} = x$$
$$\bar{\rho}_k = \frac{x_k}{v_k}$$
$$\bar{\rho}_n = \rho_n \forall n \neq k$$

Then we have $C(\bar{\rho}) < C(\rho)$ since $\bar{\rho}_k = \frac{x_k}{v_k} < \rho_k$ and $\bar{\rho}_n = \rho_n \forall n \neq k$.

And since $\bar{x}_k = \bar{\rho}_k v_k$, and $\bar{x}_k = x_k \le w_k(\rho_k^{\max} - \rho_k) \le w_k(\rho_k^{\max} - \bar{\rho}_k)$, then

$$\bar{x}_k = \min\{\bar{\rho}_k v_k, w_k(\rho_k^{\max} - \bar{\rho}_k)\}$$

and the constraint $\bar{q}_k = f(\bar{\rho}_k)$ is satisfied. All other constraints trivially hold.

Therefore, $(\bar{\rho}, \bar{x})$ is feasible for (SO), and (ρ, x) is non optimal. This shows that (ρ^*, x^*) is optimal only if $x_n^* = v_n \rho_n^*$ which completes the proof. $\qquad\square$

3.2 Simple characterization of social optima

As an immediate corollary of the previous Lemma, the social optimum can be computed by solving the following equivalent linear program

$$\min_{\rho, x} \sum_n L_n \rho_n$$
$$\text{subject to } \sum_n x_n = r$$
$$x_n \le v_n \rho_n$$
$$x_n \le x_n^{\max}$$

Then assuming the links are ordered by increasing free-flow latency $a_1 \le \cdots \le a_N$ the social optimum is simply given by the assignment that saturates most efficient links first. Formally, if $k_0 = \max\{k | r > \sum_{n=1}^k x_n^{\max}\}$ then the social optimal assignment x^* is given by

$$x^* = (x_1^{\max}, \ldots, x_{k_0}^{\max}, r - \sum_{n=1}^{k_0} x_n^{\max}, 0, \ldots, 0) \tag{5}$$

4 Stackelberg assignment

In order to reduce the inefficiency of the network, we assume that a fraction of the flow is centrally controlled, and we investigate possible strategies for improving the equilibria of the network.

4.1 Valid Stackelberg strategy

We consider the following problem: given a network under constant flow demand r, assume a coordinator (a central authority) has control over a fraction β of the flow: the corresponding users are compliant and willing to change their routes according to the instructions thy are given. The coordinator (who plays the role of the leader in the Stackelberg game) assigns the compliant flow βr according to a Stackelberg strategy s that is a feasible flow assignment for instance $(N, \beta r)$, i.e. s satisfies

$$s_n \leq x_n^{\max} \forall n \qquad\qquad \sum_n s_n = \beta r$$

We assume that the non compliant users (who play the role of followers in the Stackelberg game), with corresponding flow $(1 - \beta)r$, choose their routes selfishly after the Stackelberg assignment s of compliant users is revealed. This induces an assignment $(t(s), m(s))$ of the selfish flow at Nash equilibrium, and we assume that this is the best such Nash equilibrium (as discussed in section 2, there may be multiple such equilibria, but we only consider the most efficient equilibrium). We also assume that the assignment s of compliant users *is not affected* after introducing the non-compliant flow on the network. Note that the Nash assignment (t, m) depends on the Stackelberg strategy s.

To characterize the final Nash equilibrium, which we refer to as the *induced equilibrium* by strategy s, we note that the flow on link n is simply $s_n + t_n(s)$, and we have $\forall n$

$$t_n(s) > 0 \Rightarrow \forall k, l_n\left(s_n + t_n(s), m_n(s)\right) \leq l_k\left(s_k + t_k(s), m_k(s)\right)$$

and by lemma 1, all links that are used by selfish users have a common latency l_0 in the induced equilibrium, and links that are not used by the selfish flow have latency greater than l_0.

This can be summarized in the following definition of a valid Stackelberg strategy

Definition 3. *A valid Stackelberg strategy is an assignment s of the compliant flow βr that is feasible for the instance $(N, \beta r)$, and which induces a Nash assignment $(t(s), m(s))$ of the non-compliant flow that satisfies*

$$\forall n \in Supp\left(t(s)\right), l_n\left(s_n + t_n(s), m_n(s)\right) = l_0$$
$$\forall n \notin Supp\left(t(s)\right), l_n\left(s_n, m_n(s)\right) \geq l_0$$

4.2 Example of an invalid Stackelberg strategy

Note that a feasible flow assignment s of compliant flow may fail to induce a Nash assignment t that does not affect the initial compliant assignment s.

To see this, consider the following 2-link network where links have the same length $L_1 = L_2 = 1$, link 1 has greater free-flow speed and capacity than link 2 $v_1 > v_2$, $q_1^{\max} > q_2^{\max}$, and both links have same congestion wave speed $w_1 = w_2$. Now assume that the network is subject to flow demand $r = x_1^{\max} + \epsilon$ and most of the flow is compliant $\beta r = q_1^{\max}$. Consider the following Stackelberg strategy: $s = (x_1^{\max}, 0)$.

Assuming that the assignment of compliant users is not affected by introducing the non-compliant flow, we have for any assignment t of non-compliant flow, $t_1 = 0$ and $t_2 > 0$. Therefore t is not a Nash assignment since $Supp(t) = \{2\}$ and $l_2(s_2 + t_2, m_2) > l_1(s_1, 0)$ (non compliant users are forced to use less efficient link 2).

In practice, strategy s may induce a steady state equilibrium in which the compliant flow is some assignment $s' \neq s$: when non-compliant flow is introduced on the network, drivers will choose link 1 even though it is at maximum flow, since link 1 is faster $l_1(q_1(^{\max}, 0) < l_2(0, 0)$. This will result in congesting the link $\rho_1 > \rho_1^c$ and the flow on link 1 will *decrease* due to congestion. Now a fraction of the compliant

6

flow will not be able to use link 1, and will be forced to use link 2. This corresponds to a capacity drop on link 1 when non-compliant flow is introduced. Therefore, in the final equilibrium, the compliant flow will be $s' = (s_1', s_2')$ such that $l_1(s_1', 1) = l_2(s_2' + \epsilon, 0)$, link 1 will be congested in the final equilibrium, and link 2 in free flow. Note that such a strategy s is not considered as a valid Stackelberg assignment in our definition, we only consider strategies that induce assignment $s + t$ where t is a Nash assignment as described above.

5 Optimal Stackelberg strategies

In this section we solve for optimal Stackelberg assignments, i.e. Stackelberg assignments that induce Nash equilibria of optimal cost. This is described by the following optimization problem

$$\min_s C\left(s + t(s), m(s)\right)$$

where s is a *valid* Stackelberg assignment and $(t(s), m(s))$ is the non-compliant flow assignment at the equilibrium induced by s.

We define an optimal Stackelberg strategy s^* to be a solution to the optimization problem

$$s^* = \arg\min_s C\left(s + t(s), m(s)\right)$$

and a Stackelberg equilibrium to be the assignment $(s^* + t(s^*), m(s^*))$ where $(t(s^*), m(s^*))$ is the Nash equilibrium induced by optimal strategy $s*$.

We first study the 2 link case and solve analytically for the optimal Stackelberg strategy, then extend the result to the general case.

5.1 The 2-link case

5.2 General case

In this section we show the following result: the optimal Stackelberg strategy can be computed in polynomial time for parallel networks with N links. This result contrasts with the case of increasing latency functions where the optimal Stackelberg strategy is shown to be NP-hard to compute, see [1].

Roughly speaking, the optimal Stackelberg strategy in our case corresponds to assigning the compliant flow in a socially optimal way *after* assigning the non-compliant flow in a selfish way. Next we detail this rough idea by defining a candidate Stackelberg strategy $\bar{s}$ that will later be shown to be optimal.

5.2.1 A candidate Stackelberg strategy: Non-Compliant First

Let $(\bar{t}, \bar{m})$ denote the best Nash equilibrium for the instance $(N, (1 - \beta)r)$. Let $k = \max Supp(\bar{t})$. We have

$$\bar{m} = (1, \ldots, 1, \overset{k}{0}, \ldots, 0)$$

and

$$\bar{t} = \left(\hat{x}_1(k), \ldots, \hat{x}_{k-1}(k), (1 - \beta)r - \sum_{n=1}^{k-1} \hat{x}_n(k), 0, \ldots, 0\right)$$

i.e. links $\{1, \ldots, k - 1\}$ are k-congested, and link k is in free flow.

Now define Stackelberg strategy $\bar{s}$ as the optimal assignment of compliant flow βr that induces equilibrium $(\bar{t}, \bar{m})$. It is easy to see that $\bar{s}$ is simply given by assigning the compliant flow to remaining links $\{k, k+1, \ldots, N\}$ successively, each up to maximum capacity. The strategy $\bar{s}$ will assign $x_k^{\max} - t_k$ on link k, then $x_{k+1}^{\max}$ on link $k + 1$, $x_{k+2}^{\max}$ on link $k + 2$ and so on. Let $l = \min\{n | \beta r - (\sum_{n=k}^{l-1} x_n^{\max} - t_k) \geq 0\}$ be the least efficient link used by the Stackelberg assignment. Then $\bar{s}$ is given by

$$\bar{s} = \left(0, \ldots, \overset{k-1}{0}, x_k^{\max} - \bar{t}_k, x_{k+1}^{\max}, \ldots, x_{l-1}^{\max}, \beta r - \left(\sum_{n=k}^{l-1} x_n^{\max} - t_k \right), \overset{l+1}{0}, \ldots, 0 \right)$$

Equivalently, the total assignment $\bar{x} = \bar{s} + \bar{t}$ is given by

$$\bar{x} = \left(\hat{x}_1(k), \ldots, \hat{x}_{k-1}(k), x_k^{\max}, x_{k+1}^{\max}, \ldots, x_{l-1}^{\max}, r - \sum_{n=1}^{k-1} \hat{x}_n(k) - \sum_{n=k}^{l-1} x_n^{\max}, \overset{l+1}{0}, \ldots, 0 \right)$$

Next we show that strategy $\bar{s}$ is indeed an optimal Stackelberg strategy.

5.2.2 The Non-Compliant First strategy is optimal

We first introduce a definition that will be useful in proving the main result. Consider a network under feasible flow assignment (x, m). A link n is said to be i-congested $(i \geq n + 1)$ if n is congested $(m_n = 1)$ and its latency is at least a_i.

Definition 4. *Link n is i-congested $(i \geq n + 1)$ under assignment (x, m) if*

$$m_n = 1$$
$$l_n(x_n, m_n) \geq a_i$$

which is equivalent to

$$m_n = 1$$
$$x_n \leq \hat{x}_n(i)$$

Note that in a network under Nash assignment (t, m), if $i = \max Supp(t)$, then all links $n \in \{1, \ldots, i-1\}$ are i-congested.

Theorem 1. *$\bar{s}$ is an optimal Stackelberg strategy.*

Proof: Let s be a valid Stackelberg strategy and (t, m) be the best induced Nash equilibrium for the non-compliant flow. We will show that

$$C(x, m) \geq C(\bar{x}, \bar{m})$$

where $x = s + t$ and $\bar{x} = \bar{s} + \bar{t}$.

The proof proceeds as follows: we first show that links $\{1, \ldots, l-1\}$ are more congested under assignment (x, m) than under $(\bar{x}, \bar{m})$, in the following sense: these links have worse latency

$$l_n(x_n, m_n) \geq l_n(\bar{x}_n, \bar{m}_n)$$

and hold less flow

$$x_n \leq \bar{x}_n$$

Then we conclude by lower bounding the cost $C(x, m)$.

Let $k' = \max Supp(t)$ be the link with largest free-flow latency used by the non-compliant flow t. We begin by proving the following Lemma

Lemma 3. *Link k' is in free-flow.*

Lemma 4. *$k' \geq k$*

Proof: First note that $s + t$ restricted to $Supp(t)$ is an assignment at Nash equilibrium. Then since link k' is in free-flow and $k' \in Supp(t)$, the cost of this Nash equilibrium is simply $r'a_{k'}$ where $r' = \sum_{n \in Supp(t)} s_n + t_n$ is the total flow restricted to $Supp(t)$.

We have $\bar{t}$ is the best Nash equilibrium of flow $(1 - \beta)r$ with cost $(1 - \beta)ra_k$, and $(1 - \beta)r = \sum_{n \in Supp(t)} t_n \leq \sum_{n \in Supp(t)} s_n + t_n = r'$. Then since the average cost of the best Nash equilibrium is increasing in the total flow, we have

$$a_{k'} \geq \frac{r'a_{k'}}{r'} \geq \frac{(1 - \beta)ra_k}{(1 - \beta)r}$$

Therefore we must have $a_{k'} \geq a_k$, i.e. $k' \geq k$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Using the lemma, we can now show that links $\{1, \ldots, l - 1\}$ are more congested under assignment (x, m) than candidate assignment $(\bar{x}, \bar{m})$.

We have $\forall n \in \{1, \ldots, k' - 1\}$, n is k'-congested under assignment (x, m), and $\forall n \in \{1, \ldots, k - 1\}$, n is k-congested under assignment $(\bar{x}, \bar{m})$. Thus using the fact that $k' \geq k$

$$l_n(x_n, m_n) \geq a_{k'} \geq a_k = l_n(\bar{x}_n, \bar{m}_n) \qquad\qquad \forall n \in \{1, \ldots, k - 1\}$$
$$x_n \leq \hat{x}_n(k') \leq \hat{x}_n(k) = \bar{x}_n \qquad\qquad \forall n \in \{1, \ldots, k - 1\}$$

and we have $\forall n \in \{k, \ldots, l - 1\}$, n is in free flow and at maximum capacity under assignment $(\bar{x}, \bar{m})$ (i.e. $\bar{x}_n = q_n^{\max}$ and $l_n(\bar{x}_n) = a_n$). Thus

$$l_n(x_n, m_n) \geq a_n = l_n(\bar{x}_n, \bar{m}_n) \qquad\qquad \forall n \in \{k, \ldots, l - 1\}$$
$$x_n \leq x_n^{\max} = \bar{x}_n \qquad\qquad \forall n \in \{k, \ldots, l - 1\}$$

Therefore we have

$$l_n(x_n, m_n) \geq l_n(\bar{x}_n, \bar{m}_n) \qquad\qquad \forall n \in \{1, \ldots, l - 1\} \qquad\qquad (6)$$
$$x_n \leq \bar{x}_n \qquad\qquad \forall n \in \{1, \ldots, l - 1\} \qquad\qquad (7)$$

Note that $\forall n \in \{1, \ldots, k\}\ l_n(\bar{x}_n, \bar{m}_n) = a_k \leq a_l$, and $\forall n \in \{k, \ldots, l - 1\}\ l_n(\bar{x}_n, \bar{m}_n) = a_n \leq a_l$, thus we have

$$l_n(\bar{x}_n, \bar{m}_n) \leq a_l \qquad\qquad \forall n \in \{1, \ldots, l - 1\} \qquad\qquad (8)$$

Also note that links $n \in \{l, \ldots, N\}$ have latency at least a_n (the latency on a link is always greater than the free-flow latency) and $a_n \geq a_l$, thus

$$l_n(x_n, m_n) \geq a_l \qquad\qquad \forall n \in \{l, \ldots, N\} \qquad\qquad (9)$$

We can now lower-bound the cost of the assignment (x, m) where $x = s + t$ and (t, m) is the best Nash equilibrium induced by s.

9

$$C(x, m) = \sum_{n=1}^{l-1} x_n l_n(x_n, m_n) + \sum_{n=l}^{N} x_n l_n(x_n, m_n)$$

$$\geq \sum_{n=1}^{l-1} x_n l_n(\bar{x}_n, \bar{m}_n) + \sum_{n=l}^{N} x_n a_l \qquad \text{using 6 and 9}$$

$$= \sum_{n=1}^{l-1} \bar{x}_n l_n(\bar{x}_n, \bar{m}_n) + \sum_{n=1}^{l-1} (x_n - \bar{x}_n) l_n(\bar{x}_n, \bar{m}_n) + \sum_{n=l}^{N} x_n a_l$$

$$\geq \sum_{n=1}^{l-1} \bar{x}_n l_n(\bar{x}_n, \bar{m}_n) + \sum_{n=1}^{l-1} (x_n - \bar{x}_n) a_l + \sum_{n=l}^{N} x_n a_l \qquad \text{using 7 and 8}$$

$$= \sum_{n=1}^{l-1} \bar{x}_n l_n(\bar{x}_n, \bar{m}_n) + \left(r - \sum_{n=1}^{l-1} \bar{x}_n \right) a_l$$

$$= \sum_{n=1}^{l-1} \bar{x}_n l_n(\bar{x}_n, \bar{m}_n) + \bar{x}_l a_l \qquad \text{using } r = \sum_{n=1}^{l} \bar{x}_n \text{ since } Supp(\bar{x}) = \{1, \ldots, l\}$$

$$= C(\bar{x}, \bar{m})$$

□

References

[1] T. Roughgarden. Stackelberg Scheduling Strategies. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing, 2001*